



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/822,454	04/12/2004	Erez Haba	MSFTP641US	2824
27195 7590 04/16/2009 AMIN, TUROCY & CALVIN, LLP 127 Public Square 57th Floor, Key Tower CLEVELAND, OH 44114				
EXAMINER WANG, BEN C				
ART UNIT 2192		PAPER NUMBER		
NOTIFICATION DATE 04/16/2009		DELIVERY MODE ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

Office Action Summary

Application No.

10/822,454

Applicant(s)

HABA ET AL.

Examiner

BEN C. WANG

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 14 January 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-5, 11, 12, 14, 17, 18, 20, 22, 23 and 25-27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-5, 11-12, 14, 17-18, 20, 22-23, and 25-27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. Applicant's amendment dated January 14, 2009, responding to the Office action mailed December 10, 2008 provided in the rejection of claims 1-5, 11-12, 14, 17-18, 20, 22-23, and 25-27, wherein claims 1, 14, 17, and 23 have been amended.

Claims 1-5, 11, 12, 14, 17, 18, 20, 22, 23, and 25-27 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims rejection have been fully considered but are moot in view of the new grounds of rejection – see *Dimitriadis* - art of record, as applied hereto.

2. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-5, 11, 12, 14, 17, 18, 20, 22, 23, and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Blackwell (Pub. No. US 2005/0166094 A1) (hereinafter 'Blackwell') in view of Mandava (Pat. No. US 7,203,928 B2) (hereinafter 'Mandava-2') and Dimitris Dimitriadis, (*DOM Test Suite Methodology Report, National Institute of Standards and Technology, February 2004*) (hereinafter 'Dimitriadis' - art of record)

4. **As to claim 1** (Currently Amended), Blackwell discloses an application test management system comprising:
a computer readable storage medium having stored thereon the following components executable by a processor:

- a version component that monitors source under test components and test components for changes (e.g., [0025] – Configuration management (CM) tools are known to prepare systems for automated testing, such as managing documents and data associated with the testing ... CM tools can preserve the integrity of the scripted procedures, track versions of software, and allow scripts

to be reused and tracked ... A checkin/checkout development model is included, with versioning of directories, subdirectories and all file system objects with a versioned object database ...);

- a test catalog that provides a repository for a collection of test case files, test cases, and test variations, and namespace metadata, and is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion (e.g., [0142] – Tests cases can be organized using Test Case Catalog ... test case name or a test case descriptor ... other fields pertaining to the test case; [0144] – Information regarding test cases may be entered in a Test Case Catalog Maintenance Form ... The Test Case Maintenance Form provides a means to receive and display information about any single test case ...; [0145] - [0149]); and
- wherein the test case file component generates test results that are tagged with the versions of the source under test components and saved to a data store for historical analysis, the test results and version component are all version tagged data and dependent on the versions of the software under test (e.g., Fig. 13, element of 374 – Administrative – Test Results; Verification Report; [0240 - ... Test results are generated and compared against know-good values ... Results are, automatically compared and verified against know good results ...);

Further, Blackwell discloses a tool to assist software testing that can take advantage of the data in a Traceability Matrix to provide automated roadmaps of relationships between affected test cases in order to more efficiently select test cases needed for

proper testing following a change in one or more software modules or other system components (e.g., [0017]), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Result*, Mandava-2 discloses:

- a test case file component that is continuously modified such that new features are added and/or removed to test changes in the source under test components (e.g., Col. 3, Lines 17-31 - ... Source code for each of the plurality of test cases includes a plurality of embedded reporter code ...; Col. 8, Lines 1-16 - ... the embedded reporter codes are defined after a method call or any point in the application ...; Lines 52-67 - ... defines a plurality of embedded reporter codes in the source code of the test suite ... the embedded reporter codes are application program interfaces (APIs) ...; NOTE: there is a binding between test cases and their associated source code)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the Blackwell's system to further provide other limitations stated above in the Blackwell system.

The motivation is that it would further enhance the Blackwell's system by taking, advancing and/or incorporating Mandava-2 system which offers significant advantages that provide consistent and uniform test results; and the generated results are substantially accurate as once suggested by Mandava-2 (e.g., Col. 20, Lines 40-60)

Furthermore, Mandava-2 discloses a method for maintaining standardized test results (e.g., Abstract) but Blackwell and Mandava-2 do not explicitly disclose other limitations stated below.

However, in an analogous art of *DOM Test Suite Methodology Report*, Dimitris discloses:

- wherein the test case file component receives version data from the version component and stored the version data to an XML document, the XML document is transformed utilizing XSLT to enable a user to view at least one of exception patterns, trends, productivity, and success rates (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

- a test case file component that includes metadata associated with test components and source under test components received from the version component that indicates relationships between versions of source under test components and versions of test cases, the test case file component (e.g., P. 8, Step 3 – Generate the relevant language versions of the tests (... ECMAScript and Java) using XSLT transforms for limiting ambiguity ...; P. 9, 5th Para - ... Provide links to the test cases themselves ... Links were provided to the test storage space and to the local checked out version of the test suit. These tests could be browsed from a table built by the building tool to allow for test case authors to look at the actual test ... pointing to files in a file system ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 - Test Case Table and Traceability; Sec. 2.5.5 – Data Management, 1st Para - ... A Concurrent Versioning System (CVS) is used for test case versioning and storage ...); and
- a build drop component that comprises an executable version of the software under test and includes changed data from the version component (e.g., P. 9, 1st Para - ... The builds are the end product of build tool's gathering tests, validating and transforming them ...; Sec. 2.6 – DOM TS Methodology, last 2nd Para - ... Any metadata such as version, part of the specification, and so on, is set on the test representation level, allowing for the build tools to configure, based on this metadata ...);

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dimitris into the Blackwell-

Mandava-2's system to further provide other limitations stated above in the Blackwell-Mandava-2 system.

The motivation is that it would further enhance the Blackwell-Mandava-2's system by taking, advancing and/or incorporating Dimitris system which offers significant advantages that easier to write conformant implementations if it is possible to test the implementation during development; and less time needed to write test cases if tests can be derived or otherwise transformed form a basic set of test case descriptions; and conformance claims, oftenly being used not only as technical information but also as marketing tool, are more easily resolved if the test suite is complete and accurate as once suggested by Dimitris (e.g., Sec. 2.2 Reasons for Designing the DOM TS)

5. **As to claim 2** (original) (incorporating the rejection in claim 1), Dimitris discloses the system wherein the test case file component includes a pointer to the source under test (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write

the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

6. **As to claim 3** (original) (incorporating the rejection in claim 1), Dimitris discloses the system wherein the test case file component includes a pointer to requirement for test data (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

7. **As to claim 4** (original) (incorporating the rejection in claim 1), Dimitris discloses the system wherein the test case file component includes a pointer to requirement and/or configuration under test data (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

8. **As to claim 5** (original) (incorporating the rejection in claim 1), Dimitris discloses the system wherein the test case file component includes a pointer to a test case component (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce

a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

9. **As to claim 11** (Currently Amended) (incorporating the rejection in claim 1), Blackwell discloses the system wherein the test case component specified in the test case file component is loaded into the test catalog (e.g., [0142] – Tests cases can be organized using a Test Case Catalog ...)

10. **As to claim 12** (original) (incorporating the rejection in claim 11), Mandava-2 discloses the system wherein a test execution component executes the test case on the software under test and generates test results (e.g., Fig. 1, element 102 – target application; Col. 7, Lines 51-53 – a plurality of applications 104a – 104n for execution by a target application; Fig. 1, element 114 – dynamic XML file)

11. **As to claim 14** (Currently Amended), Blackwell discloses a test management system comprising:

a computer readable storage medium having stored thereon the following components executable by a processor:

- a means for generating test results that are tagged with test version data in relation to the version of software under test, the test results and test version data are all version tagged data and dependent on the versions of the software under test (e.g., Fig. 13, element of 374 – Administrative – Test Results; Verification Report; [0240 - ... Test results are generated and compared against know-good values ... Results are, automatically compared and verified against know good results ...])

Further, Blackwell discloses a tool to assist software testing that can take advantage of the data in a Traceability Matrix to provide automated roadmaps of relationships between affected test cases in order to more efficiently select test cases needed for proper testing following a change in one or more software modules or other system components (e.g., [0017]), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Result*, Mandava-2 discloses:

- a means for continuously modifying test data such that new features are added and/or removed to test version changes to the software under test (e.g., Col. 3, Lines 17-31 - ... Source code for each of the plurality of test cases includes a plurality of embedded reporter code ...; Col. 8, Lines 1-16 –

... the embedded reporter codes are defined after a method call or any point in the application ...; Lines 52-67 - ... defines a plurality of embedded reporter codes in the source code of the test suite ... the embedded reporter codes are application program interfaces (APIs) ...; NOTE: there is a binding between test cases and their associated source code)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the Blackwell's system to further provide other limitations stated above in the Blackwell system.

The motivation is that it would further enhance the Blackwell's system by taking, advancing and/or incorporating Mandava-2 system which offers significant advantages that provide consistent and uniform test results; and the generated results are substantially accurate as once suggested by Mandava-2 (e.g., Col. 20, Lines 40-60)

Furthermore, Mandava-2 discloses a method for maintaining standardized test results (e.g., Abstract) but Blackwell and Mandava-2 do not explicitly disclose other limitations stated below.

However, in an analogous art of *DOM Test Suite Methodology Report*, Dimitris discloses:

- a means for maintaining fine-grained track of a test's relation to a version of software under test (e.g., P. 8, Step 3 – Generate the relevant language versions of the tests (... ECMAScript and Java) using XSLT transforms for limiting ambiguity ...; P. 9, 5th Para - ... Provide links to the test cases themselves ... Links were provided to the test storage space and to the local

- checked out version of the test suit. These tests could be browsed from a table built by the building tool to allow for test case authors to look at the actual test ... pointing to files in a file system ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 - Test Case Table and Traceability; Sec. 2.5.5 – Data Management, 1st Para - ... A Concurrent Versioning System (CVS) is used for test case versioning and storage ...); and
- a means for querying test data to facilitate generation of test management reports wherein the means for maintaining fine-grained track of a test's relation to a version of software under test includes persisting software version information and related test information to an XML file, and wherein the XML file is transformed utilizing XSLT to enable a user to view at least one of exception patterns, trends, productivity, and success rates (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL

is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dimitris into the Blackwell-Mandava-2's system to further provide other limitations stated above in the Blackwell-Mandava-2 system.

The motivation is that it would further enhance the Blackwell-Mandava-2's system by taking, advancing and/or incorporating Dimitris system which offers significant advantages that easier to write conformant implementations if it is possible to test the implementation during development; and less time needed to write test cases if tests can be derived or otherwise transformed form a basic set of test case descriptions; and conformance claims, oftenly being used not only as technical information but also as marketing tool, are more easily resolved if the test suite is complete and accurate as once suggested by Dimitris (e.g., Sec. 2.2 Reasons for Designing the DOM TS)

12. **As to claim 17** (Currently Amended), Blackwell discloses a test management methodology comprising:

- persisting the metadata to a markup language file versioned with test assets and source code (e.g., [0294]; Table 1; [0297]; Table 3); and

- generating test results that are tagged with test version information in relation to software code version under test, the test results and test version information are all version tagged data and dependent on the versions of the software code under test (e.g., Fig. 13, element of 374 – Administrative – Test Results; Verification Report; [0240 - ... Test results are generated and compared against know-good values ... Results are, automatically compared and verified against know good results ...]);

Further, Blackwell discloses a tool to assist software testing that can take advantage of the data in a Traceability Matrix to provide automated roadmaps of relationships between affected test cases in order to more efficiently select test cases needed for proper testing following a change in one or more software modules or other system components (e.g., [0017]), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Result*, Mandava-2 discloses:

- continuously modifying test information such that new features are added and/or removed to test version changes to the software code under test, wherein the file is an XML file (e.g., Col. 3, Lines 17-31 - ... Source code for each of the plurality of test cases includes a plurality of embedded reporter code ...; Col. 8, Lines 1-16 - ... the embedded reporter codes are defined after a method call or any point in the application ...; Lines 52-67 - ... defines a plurality of embedded reporter codes in the source code of the test suite ... the embedded reporter codes are

application program interfaces (APIs) ...; NOTE: there is a binding between test cases and their associated source code)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the Blackwell's system to further provide other limitations stated above in the Blackwell system.

The motivation is that it would further enhance the Blackwell's system by taking, advancing and/or incorporating Mandava-2 system which offers significant advantages that provide consistent and uniform test results; and the generated results are substantially accurate as once suggested by Mandava-2 (e.g., Col. 20, Lines 40-60)

Furthermore, Mandava-2 discloses a method for maintaining standardized test results (e.g., Abstract) but Blackwell and Mandava-2 do not explicitly disclose other limitations stated below.

However, in an analogous art of *DOM Test Suite Methodology Report*, Dimitris discloses:

- retrieving metadata regarding test version information in relation to software code version under test (e.g., P. 8, Step 3 – Generate the relevant language versions of the tests (... ECMAScript and Java) using XSLT transforms for limiting ambiguity ...; P. 9, 5th Para - ... Provide links to the test cases themselves ... Links were provided to the test storage space and to the local checked out version of the test suit. These tests could be browsed from a table built by the building tool to allow for test case authors to look at the actual test ... pointing to files in a file system ...; Fig. 3 – Generating the

Metadata Table; Sec. 2.5.4 - Test Case Table and Traceability; Sec. 2.5.5 – Data Management, 1st Para - ... A Concurrent Versioning System (CVS) is used for test case versioning and storage ...); and

- transforming the XML file utilizing XSLT to enable a user to view at least one of exception patterns, trends, productivity, and success rates and management operations including at least one of selection, query, reporting, suit composition, and scheduling (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dimitris into the Blackwell-Mandava-2's system to further provide other limitations stated above in the Blackwell-Mandava-2 system.

The motivation is that it would further enhance the Blackwell-Mandava-2's system by taking, advancing and/or incorporating Dimitris system which offers significant advantages that easier to write conformant implementations if it is possible to test the implementation during development; and less time needed to write test cases if tests can be derived or otherwise transformed from a basic set of test case descriptions; and conformance claims, oftenly being used not only as technical information but also as marketing tool, are more easily resolved if the test suite is complete and accurate as once suggested by Dimitris (e.g., Sec. 2.2 Reasons for Designing the DOM TS)

13. **As to claim 18** (Previously Amended) (incorporating the rejection in claim 17), Dimitris discloses the method wherein version information is retrieved from a version component that monitors changes to source code versions and test versions (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to

evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

14. **As to claim 20** (Currently Amended) (incorporating the rejection in claim 17), Blackwell discloses the method wherein the file comprises a pointer to at least one of a source under test (e.g.), and Mandava-2 discloses requirement under test, and configuration under test (e.g., Col. 4, Lines 45-47 - ... including a respective description entry explaining the function of each test case or test)

15. **As to claim 22** (original) (incorporating the rejection in claim 17), please refer to claim 17 as set forth above accordingly.

16. **As to claim 23** (Currently Amended), Blackwell discloses a testing methodology comprising:

- loading a test case in accordance with a test ease file stored in a source file (e.g., Fig. 12, element 336 - Select Cases for Testing);

- executing the test case on a source under test (e.g., Fig. 12, element 340 – Run Test Case Scripts);
- generating test results, wherein the test results are version tagged to indicate the relationships between test results, version of the test case, and version of the source code under test (e.g., Fig. 13, element of 374 – Administrative – Test Results; Verification Report; [0240 - ... Test results are generated and compared against know-good values ... Results are, automatically compared and verified against know good results ...); and
- providing a repository for a collection of test case files, test cases, test variations, and namespace metadata, wherein the repository is constructed from aggregation of individual test case files which relate to each other in a hierarchical fashion (e.g., [0142] – Tests cases can be organized using Test Case Catalog ... test case name or a test case descriptor ... other fields pertaining to the test case; [0144] – Information regarding test cases may be entered in a Test Case Catalog Maintenance Form ... The Test Case Maintenance Form provides a means to receive and display information about any single test case ...; [0145] - [0149])

Further, Blackwell discloses a tool to assist software testing that can take advantage of the data in a Traceability Matrix to provide automated roadmaps of relationships between affected test cases in order to more efficiently select test cases needed for proper testing following a change in one or more software modules or other system components (e.g., [0017]), but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Method and System for Generating and Maintaining Uniform Test Result*, Mandava-2 discloses:

- employing XSLT to facilitate management operations including at least one of query, reporting, suite composition and scheduling (e.g., Fig. 1, elements 110 – Static XML File; 118 – Dynamic XML Results File; 120 – XSLT Interface; Col. 8, Lines 17-25 - ... the uniform results are generated ... stored to a dynamic XML file ... The uniform results in the dynamic XML can be viewed by a user using a, Extensible Stylesheet Language (XSLT) Stylesheet interface; Lines 33-51 - ... while developing the source code for the test suite, the test suite developer also create a companion static XML file ... The static XML file is written in accordance with a selected document type definition ("DTD") and is configured to provide an XML DTD for defining the static XML file ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Mandava-2 into the Blackwell's system to further provide other limitations stated above in the Blackwell system.

The motivation is that it would further enhance the Blackwell's system by taking, advancing and/or incorporating Mandava-2 system which offers significant advantages that provide consistent and uniform test results; and the generated results are substantially accurate as once suggested by Mandava-2 (e.g., Col. 20, Lines 40-60)

Furthermore, Mandava-2 discloses a method for maintaining standardized test results (e.g., Abstract) but Blackwell and Mandava-2 do not explicitly disclose other limitations stated below.

However, in an analogous art of *DOM Test Suite Methodology Report*, Dimitris discloses:

- continuously modifying test information such that new features are added and/or removed to test version changes to the source code under test (e.g., P. 8, Step 3 – Generate the relevant language versions of the tests (... ECMAScript and Java) using XSLT transforms for limiting ambiguity ...; P. 9, 5th Para - ... Provide links to the test cases themselves ... Links were provided to the test storage space and to the local checked out version of the test suit. These tests could be browsed from a table built by the building tool to allow for test case authors to look at the actual test ... pointing to files in a file system ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 - Test Case Table and Traceability; Sec. 2.5.5 – Data Management, 1st Para - ... A Concurrent Versioning System (CVS) is used for test case versioning and storage ...; P. 9, 1st Para - ... The builds are the end product of build tool's gathering tests, validating and transforming them ...; Sec. 2.6 – DOM TS Methodology, last 2nd Para - ... Any metadata such as version, part of the specification, and so on, is set on the test representation level, allowing for the build tools to configure, based on this metadata ...);
- generating test results, wherein the test results are version tagged to indicate the relationships between test results, version of the test case, and version of the source code under test (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating

the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth); and

- saving test results to an XML file, the XML file stores metadata associated with the test case and source code and contains at least one of pointers to the source under test, requirements under test and configuration under test (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to

the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures and so forth)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Dimitris into the Blackwell-Mandava-2's system to further provide other limitations stated above in the Blackwell-Mandava-2 system.

The motivation is that it would further enhance the Blackwell-Mandava-2's system by taking, advancing and/or incorporating Dimitris system which offers significant advantages that easier to write conformant implementations if it is possible to test the implementation during development; and less time needed to write test cases if tests can be derived or otherwise transformed form a basic set of test case descriptions; and conformance claims, oftenly being used not only as technical information but also as marketing tool, are more easily resolved if the test suite is complete and accurate as once suggested by Dimitris (e.g., Sec. 2.2 Reasons for Designing the DOM TS)

17. **As to claim 25** (original) (incorporating the rejection in claim 23), Mandava-2 discloses the method further comprising publishing the test results to an enterprise data store (e.g., Fig. 1, element 116; Col. 8, Lines 17-26 - ... the uniform results are stored to storage 116 in a dynamic XML result file ...)

18. **As to claim 26** (original) (incorporating the rejection in claim 23), Dimitris discloses the method wherein the version tags indicate the version of the source under test and the version of the test (e.g., P. 12, 2nd Para – Example of the Code Generation Technique – The XML document below is the test description for getting the value of an attribute node in an external XML document ...; Fig. 3 – Generating the Metadata Table; Sec. 2.5.4 – Test Case Table and Traceability, 1st Para – ... XSLT style-sheets are used to produce a table which contains a pointer to each test, part of specification tested by the test ... as well as a pointer to both the JavaScript and Java version of the test ... This table can be used to evaluate the coverage of the specification in terms of tests written for its various modules/aspects and provides traceability from the test to the relevant part of the specification; Fig. 4 – Overview of DOM TS; P. 17, item #2 – The TCDL is used to write the actual test descriptions. The build process stipulates that only valid test ... can be used to generate code; Sec. 2.6 – DOM TS Methodology, last Para - ... the latter (test suite) represents coverage and reporting issues (grouping, successes/failures) and so forth)
19. **As to claim 27** (original) (incorporating the rejection in claim 23), please refer to claim 23 above, accordingly.

Conclusion

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Ben C. Wang
Examiner, Art Unit 2192

/Eric B. Kiss/
Eric B. Kiss
Primary Examiner, Art Unit 2192